Higher Order Functions

map, filter, forEach, reduce, find

తెలుగు లో
JavaScript

Have you heard the below terms?

- Functional Programming?
- First Class Functions?
- Higher Order Functions?
- Functions are Objects?
- Are map and forEach same?
- Are filter and find same?

Functional Programming

- Functional Programming is a form of programming in which you can pass functions as parameters to other functions and also return them as values.

- In functional programming, we think and code in terms of functions.

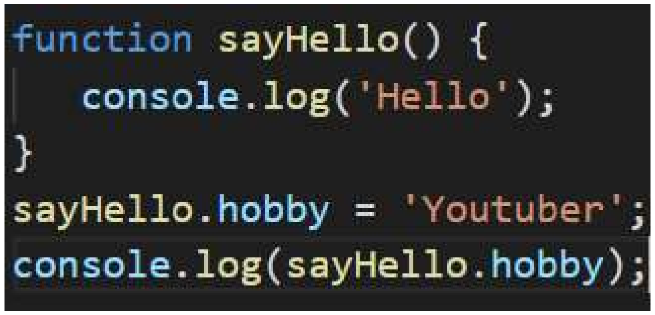
Examples: JavaScript, Haskell, Clojure, Scala, and Erlang

First Class Functions

- First Class function built with the **intention of being passed around to other functions** and will do a **specific thing**.

- Functions are objects in any Functional Programming Language.

- Do you want proof to say functions are objects in JavaScript?



```
function sayHello() {
    console.log('Hello');
}
sayHello.hobby = 'Youtuber';
console.log(sayHello.hobby);
```

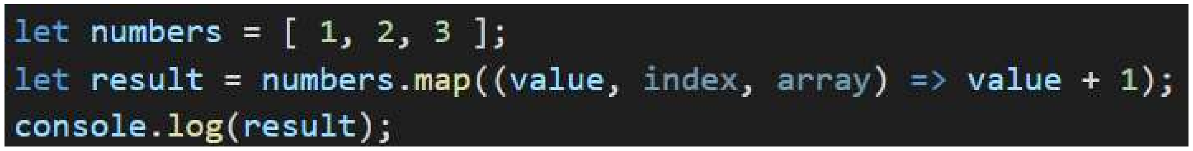
Output:

```
Youtuber
```

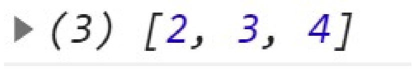
Higher Order Functions

- A Higher-Order function is a function that **receives a first class function as an argument** or **returns the first class function as output**.

Example: map, filter, reduce, forEach etc.,



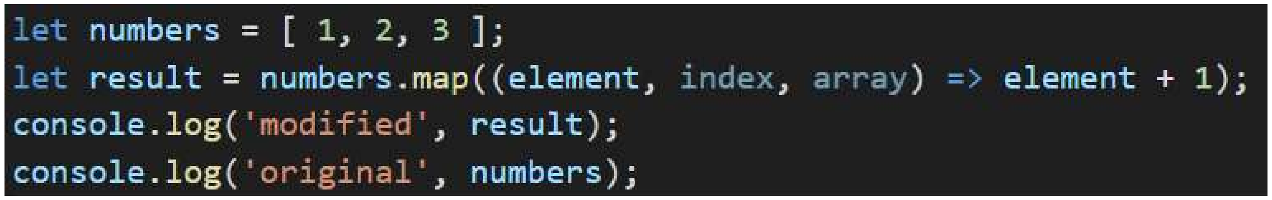
```
let numbers = [ 1, 2, 3 ];
let result = numbers.map((value, index, array) => value + 1);
console.log(result);
```



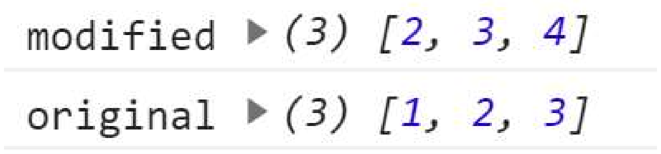
```
▶ (3) [2, 3, 4]
```

map

- The **map()** method creates new array by calling the callback function provided as an argument on every element in the input array.
- It will take every returned value from the callback function and creates a new array using those values.
- The callback function passed to the **map()** method accepts 3 arguments: **element, index and array.**



```
let numbers = [ 1, 2, 3 ];
let result = numbers.map((element, index, array) => element + 1);
console.log('modified', result);
console.log('original', numbers);
```



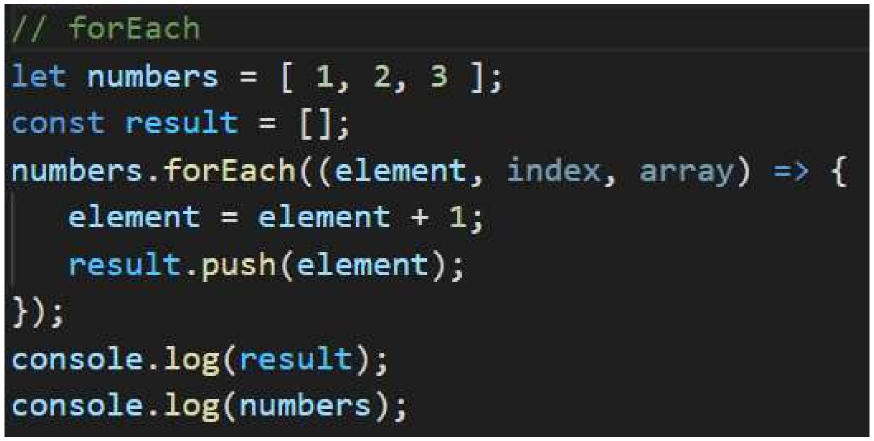
```
modified ▶ (3) [2, 3, 4]
original ▶ (3) [1, 2, 3]
```

forEach

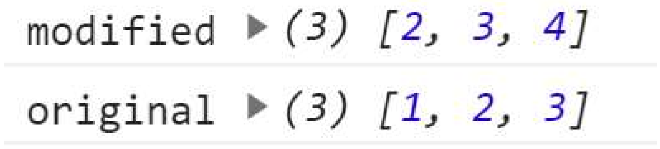
VENKATESH MOGILI
#Web Guru

➢ The **forEach()** method used to perform action for each element in the array.

➢ It accepts 3 arguments: **element, index and array.**



```
// forEach
let numbers = [ 1, 2, 3 ];
const result = [];
numbers.forEach((element, index, array) => {
    element = element + 1;
    result.push(element);
});
console.log(result);
console.log(numbers);
```



```
modified ▶(3) [2, 3, 4]
original ▶(3) [1, 2, 3]
```

Interview question

➢ What is the difference between map and forEach?

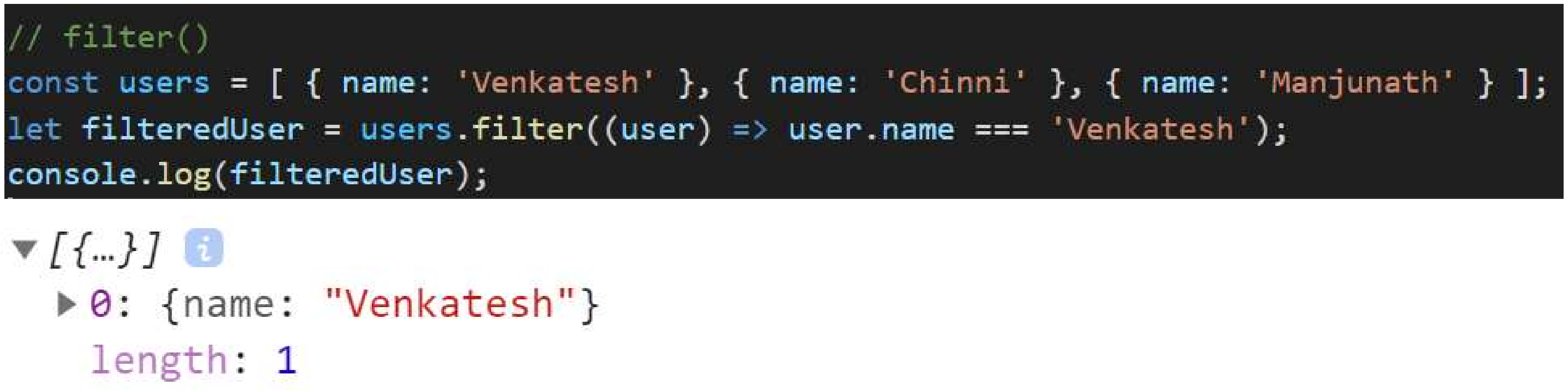
Ans:
map() returns the new array with modified values.

forEach() will perform action for each element in the array.

filter

- The **filter()** method creates a new array with all elements that pass the test provided by the callback function.

- The callback function passed to the **filter()** method accepts 3 arguments: **element, index and array.**



```
// filter()
const users = [ { name: 'Venkatesh' }, { name: 'Chinni' }, { name: 'Manjunath' } ];
let filteredUser = users.filter((user) => user.name === 'Venkatesh');
console.log(filteredUser);
```

```
▼ [{…}]
  ▶ 0: {name: "Venkatesh"}
    length: 1
```

Interview question

➢ What is the difference between filter , find and findIndex?

Ans:
filter() returns the new array which are passed the test provided by callback function.

find() returns the value of the first element in the array where predicate is true, and undefined otherwise.

findIndex() returns the index of the first element in the array where predicate is true, and -1 otherwise.

reduce

- The **reduce()** method executes the callback function on each member of the calling array which results in a single output value.

- It accepts 2 parameters: **reducer function, initial value(optional)**

- reducer function accepts 4 parameters: **accumulator, currentValue, currentIndex, sourceArray.**

- If an **initialValue** is provided, then the **accumulator** will be equal to the **initialValue** and the **currentValue** will be equal to the first element in the array.

- If no **initialValue** is provided, then the **accumulator** will be equal to the first element in the array and the **currentValue** will be equal to the second element in the array.

Summary

VENKATESH MOGILI
#Web Guru

- ✓ map, filter, reduce methods will create new array.
- ✓ map will return modified array.
- ✓ filter will return the array of values which passes the condition.
- ✓ reduce will reduce the elements to particular action which is mentioned in reducer function.
- ✓ forEach will perform action for each element in the array.
- ✓ find will return the first value of which condition matches.
- ✓ findIndex will return the first value’s index of which condition matches.

SUBSCRIBE

Don't forget to Share and Subscribe

LIKE SHARE COMMENT

SUBSCRIBE

Thank you for watching

Subscribe

and

Stay Tuned

For more updates…

VENKATESH MOGILI

#Web Guru